UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/650,245	08/27/2003	Wolfgang Grieskamp	3382-64897	7189

26119   7590   12/11/2008
KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER
DAO, THUY CHAN

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
12/11/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/650,245	GRIESKAMP ET AL.
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 October 2008*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-8,10,11,17,19 and 30-37* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-8,10,11,17,19 and 30-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 August 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

U.S. Patent and Trademark Office

PTOL-326 (Rev. 08-06)　　　　　　　Office Action Summary　　　　　Part of Paper No./Mail Date 20081206

DETAILED ACTION

1. This action is responsive to the amendment filed on October 10, 2008.

2. Claims 1, 3-8, 10, 11, 17, 19, and 30-37 have been examined.

Response to Amendments

3. In the instant amendment, claims 1, 3-6, and 30-31 have been amended.

4. The objection to the specification and claim 3 is withdrawn in view of Applicant's amendments.

5. The 35 USC §101 rejection over claims 1, 3-6, 17, and 30-31 is withdrawn in view of Applicant's amendments.

Response to Arguments

6. Applicants' arguments have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections – 35 USC §102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1, 3-8, 10, 11, 17, 19, and 30-37 are rejected under 35 U.S.C. 102(b) as being anticipated by US Patent No. 5,220,669 to Baum et al. (art made of record, hereafter "Baum").

Claim 1:

Baum discloses *a computer readable storage medium having computer-executable software code carried thereon for executing on a computing device for saving state for a semantically accessible state binding method, comprising:*

a data structure comprising: a first state frame including a representation of a first state of an executing program (e.g., FIG. 9, frame 36, col.15: 6 – col.16: 16), *and*

a second state frame including a representation of a second state of the executing program comprising state changes made by the executing program after the first state frame is created (e.g., FIG. 9, frame 37, col.16: 9-41) *and*

the second state frame includes a pointer back to the first state frame (e.g., FIG. 9, frame 37 has a backward pointer pointing back to frame 36, col.16: 17-47);

a third state frame including a representation of state changes made by the executing program after a fork method creates the third state frame (e.g., FIG. 9, frame 38, which may be created by a "branch and stack" instruction, col.16: 35-48; col.18: 62 – col.19: 34), *and*

wherein the third state frame is empty when created by the fork method (e.g., FIG. 10, a new linkage stack LS-state entry is empty when newly created, col.16: 41-52);

wherein the executing program saves a first state of the program in the first state frame as a semantically accessible first state object (e.g., FIG. 9, for example "frame36.header", "frame36.trailer", and "frame36.state", col.16: 9-41),

saves a second state of the program in the second state frame as a semantically accessible second state object (e.g., FIG. 9, for example "frame37.header", "frame37.trailer", and "frame37.state", col.16: 9-41), *and*

then returns to the first state of the program by using the first state object (e.g., FIG. 9, for example "frame37.backward_pointer = *frame36.trailer", i.e., assigning the address of frame36.trailer (first state object) to the backward pointer of frame37, so that the system can return to frame 36 (first state), col.15: 61 – col.16: 48).

the third state frame includes a pointer back to the second frame (e.g., FIG. 9, frame 38 has backward pointer pointing back to frame 37, col.16: 17-41)

wherein, after the fork method creates the third state frame, value of a variable of the executing program can be accessed by checking, in backwards order that the frames were created, for the value of the variable in the respective frames (e.g.,

FIG. 9, after the "branch and stack" instruction (the fork method) creates frame 38 (third state frame), values of frame 37 or 36 can be accessed by using backward pointers, col.17: 9-52; col.18: 62 – col.19: 34).

Claim 3:

The rejection of claim 1 is incorporated. Baum discloses *a fourth state frame which includes changes made by the executing program after the fork method creates the third state frame* (e.g., col.4: 66 – col.5: 55; col.12: 38-50) *and*

after a set method returns the executing program to the state of the second state frame, and wherein the fourth state frame includes a pointer back to the second state frame (e.g., col.6: 11 – col.7: 30; col.14: 39-58).

Claim 4:

The rejection of claim 3 is incorporated. Baum discloses *a joined state frame including a combination of state changes in the third and fourth state frames* (e.g., col.15: 60-48; col.19: 47-62).

Claim 5:

The rejection of claim 3 is incorporated. Baum discloses *a first thread of the executing program makes state changes copied in the second state frame* (e.g., col.7: 31 – col.8: 16)*, and*

a second thread of the executing program makes state changes copied into the third state frame (e.g., col.9: 12-59).

Claim 6:

The rejection of claim 1 is incorporated. Baum discloses *the second state frame includes unchanged state read from the first state frame* (e.g., col.19: 47-62; col.21: 12-53).

Claim 7:

Baum discloses *a computerized method comprising:*

receiving via an application programming interface a request to create a state save (e.g., FIG. 9, frame 36, col.15: 6 – col.16: 16; frame 38, which may be created by a "branch and stack" instruction, col.16: 35-48; col.18: 62 – col.19: 34)*;*

in response to the request, saving a first representation of a state of an executing program comprising copying state of the program required to return to the moment the state was saved as a first state frame (e.g., FIG. 9, frame 37 has a backward pointer pointing back to frame 36, col.16: 17-47)*;*

creating a blank state frame with a backward link to the first state frame as a current state frame (e.g., FIG. 10, a new linkage stack LS-state entry is empty when newly created, col.16: 41-52; FIG. 9, frame 38 has backward pointer pointing back to frame 37, col.16: 17-41)*;*

maintaining a second representation of subsequent state comprising changes made to the state of the executing program after the first representation in the current state frame (e.g., FIG. 9, frame 37 has a backward pointer pointing back to frame 36, col.16: 17-47)*;*

in response to a request for value of a variable after the request to create a state save, checking for the value of the variable in the first state frame (e.g., FIG. 9, for example "frame36.header", "frame36.trailer", and "frame36.state", col.16: 9-41)*; and*

changing the current state frame to the first state frame upon receiving a state set request at the application programming interface (e.g., FIG. 9, values of frame 37 or 36 can be accessed by using backward pointers, col.17: 9-52; col.18: 62 – col.19: 34).

Claim 8:

Baum discloses *a computer system comprising: memory and a central processing unit executing,*

a program including executable instructions and an evolving present state (e.g., FIG. 9, values of frame 37 or 36 can be accessed by using backward pointers, col.17: 9-52; col.18: 62 – col.19: 34)*;*

a first state frame comprising an initial representation of a prior evolving present state of the program (e.g., FIG. 9, frame 36, col.15: 6 – col.16: 16)

a state component comprising a representation of the first state frame as a programmer-accessible named object (e.g., FIG. 9, for example "frame36.header", "frame36.trailer", and "frame36.state", col.16: 9-41),

a second state frame comprising a subsequent representation of state changes made by the program since the initial representation (e.g., FIG. 9, for example "frame36.header", "frame36.trailer", and "frame36.state", col.16: 9-41),

the program including a method for returning the program state to the prior evolving present state using the state component (e.g., FIG. 9, for example "frame37.backward_pointer = *frame36.trailer", i.e., assigning the address of frame36.trailer (first state object) to the backward pointer of frame37, so that the system can return to frame 36 (first state), col.15: 61 – col.16: 48); *and*

the program including a method for locating a value updated during the prior evolving present state and not present in the second state frame by following a back pointer from the second state frame to the first state frame (e.g., FIG. 9, frame 38 has backward pointer pointing back to frame 37, col.16: 17-41)

reading location value from the first state frame and storing the location value in the second state frame (e.g., FIG. 9, frame 37, col.16: 9-41; FIG. 9, values of frame 37 or 36 can be accessed by using backward pointers, col.17: 9-52; col.18: 62 – col.19: 34).

Claim 10:

The rejection of claim 8 is incorporated. Baum discloses *the state component includes a fork method for maintaining state for a thread spawned by the program and a forked representation of state changes made by the spawned thread of the program* (e.g., col.16: 17 – col.17: 27; col.19: 47-62).

Claim 11:

The rejection of claim 10 is incorporated. Baum discloses *the state component includes a join method for joining state changes made by the forked thread back into state changes of the subsequent representation* (e.g., col.7; 31 – col.8: 16; col.19: 47-62).

Claim 17:

Claim 17 is a computer readable storage medium version, which recites the same limitations as those of claim 7, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claim, they also teach all of the limitations of claim 17.

Claim 19:

Baum discloses *a computerized method comprising computer executable instructions for performing a method comprising:*

receiving a request from a method, which takes as a parameter a state object, to create a saved state of an executing model (e.g., FIG. 9, for example "frame36.header", "frame36.trailer", and "frame36.state", col.16: 9-41; FIG. 9, for example "frame37.backward_pointer = *frame36.trailer", i.e., assigning the address of frame36.trailer (first state object) to the backward pointer of frame37, so that the system can return to frame 36 (first state), col.15: 61 – col.16: 48)*;

saving a first representation of a state of the executing model as a first state frame; saving a first representation of the state frame as the state object (e.g., FIG. 9, frame 36, col.15: 6 – col.16: 16)*;

creating a blank state frame with a backward link to the first state frame as a second state frame (e.g., FIG. 10, a new linkage stack LS-state entry is empty when newly created, col.16: 41-52)*

maintaining, in the second state frame, a second representation of state changes made by the executing model after the first representation as the state changes occur (e.g., FIG. 9, frame 37, col.16: 9-41)*; and*

reinstating the executing model state to the state of the first representation using the state object (e.g., FIG. 9, for example "frame37.backward_pointer = *frame36.trailer", i.e., assigning the address of frame36.trailer (first state object) to the backward pointer of frame37, so that the system can return to frame 36 (first state); col.15: 61 – col.16: 48; values of frame 37 or 36 can be accessed by using backward pointers, col.17: 9-52; col.18: 62 – col.19: 34).

Claim 30:

The rejection of claim 3 is incorporated. Baum discloses *the data structure further comprises a fifth state frame with a pointer to the fourth state frame, wherein the fourth state frame further comprises a reference pointer with a value indicating how many frames point back to it* (e.g., col.6: 11 – col.7: 30; col.14: 39-58)*, and*

wherein if the value of the reference pointer is one, then the executing program combines the fourth state frame with the fifth state frame (e.g., col.7: 31 – col.8: 16; col.15: 60 – col.16: 48).

Claim 31:

The rejection of claim 3 is incorporated. Baum discloses *the fourth state frame includes a pointer to the third state frame* (e.g., col.16: 9-41; col.18: 62 – col.19: 34).

Claim 32:

The rejection of claim 7 is incorporated. Baum discloses *writing the value of the variable in a cache in the current state frame* (e.g., col.16: 17-47; col.19: 47-62).

Claim 33:

The rejection of claim 32 is incorporated. Baum discloses *a threshold size wherein when the cache is greater than the threshold size, the cache is purged* (e.g., col.16: 35-48; col.21: 12-53).

Claim 34:

The rejection of claim 32 is incorporated. Baum discloses *a threshold size wherein when the cache is greater than the threshold size, the last used variable is overwritten* (e.g., col.15: 61 – col.16: 16; col.6: 11 – col.7: 30).

Claim 35:

The rejection of claim 19 is incorporated. Baum discloses:

receiving a request from a fork method which saves state of the executing program as a first state frame when invoked with the state object; receiving a request from a set method which sets the state of the executing program to the first state when invoked with the state object (e.g., col.7: 31 – col.8: 16; col.16: 17 – col.17: 27); *and*

receiving a request from a join method which joins the current state of the executing program with the first state when invoked with the state object (e.g., col.9: 12-59; col.16: 41 – col.17: 52).

Claim 36:

The rejection of claim 19 is incorporated. Baum discloses *the first frame and the second frame have a creation order* (e.g., col.16: 9-41; col.18: 62 – col.19: 34).

Claim 37:

The rejection of claim 36 is incorporated. Baum discloses *when a variable is accessed by the executing model, checking the state frames in opposite order to their creation order until an update for the variable is found* (e.g., col.14: 39-58; col.16: 17 – col.17: 27).

Conclusion

9. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/   /Tuan Q. Dam/
Examiner, Art Unit 2192   Supervisory Patent Examiner, Art Unit 2192